Counting Consistent Phylogenetic Trees is #P-complete

M. Bordewich, C.Semple and J. Talbot

Department of Mathematics and Statistics
University of Canterbury
Private Bag 4800
Christchurch, New Zealand

Report Number: UCDMS2003/11 JULY 2003

Counting Consistent Phylogenetic Trees is #P-complete

Magnus Bordewich*, Charles Semple†, John Talbot‡

18 July 2003

Abstract

Reconstructing phylogenetic trees is a fundamental task in evolutionary biology. Various algorithms exist for this purpose, many of which come under the heading of 'supertree methods'. These methods amalgamate a collection $\mathcal{P}$ of phylogenetic trees into a single parent tree. In this paper, we show that, in both the rooted and unrooted settings, counting the number of parent trees that preserve all of the ancestral relationships displayed by the phylogenetic trees in $\mathcal{P}$ is #P-complete.

1 Introduction

Phylogenetics is the reconstruction and analysis of phylogenetic (evolutionary) trees and networks based on inherited characteristics. In evolutionary biology, phylogenetic trees are used to represent the ancestral history of a collection of present-day species.

There exists a variety of methods for reconstructing phylogenetic trees depending upon the type of information being used for inference. *Supertree methods* is the collective name for reconstruction algorithms that combine a collection $\mathcal{P}$ of smaller phylogenetic trees on overlapping sets of species into a single parent tree. The resulting parent tree is called a *supertree*. Supertree methods have attracted much interest in evolutionary biology as illustrated by a recent survey paper [2] and a soon to be published book [3].

*Corresponding author. bordewic@maths.ox.ac.uk. New College and Mathematical Institute, Oxford, UK.

†c.semple@math.canterbury.ac.nz. Biomathematics Research Centre, Department of Mathematics and Statistics, University of Canterbury, Christchurch, New Zealand.

‡talbot@maths.ox.ac.uk. Merton College and Mathematical Institute, Oxford, UK.

A desirable property of any such method is that the resulting supertree preserves (if possible) all of the ancestral relationships described by the smaller phylogenetic trees. Such a supertree is said to be *consistent* with $\mathcal{P}$.

If $\mathcal{P}$ is a collection of rooted binary phylogenetic trees, then deciding whether there exists a consistent rooted binary supertree for $\mathcal{P}$ can be done in polynomial time [1]. Indeed, the associated algorithm outputs an appropriate supertree if one exists. For biologists who may want to determine the evolutionary history of up to $10,000$ species on a single tree, the efficiency of this algorithm has important practical implications. However, knowing there is at least one consistent rooted binary supertree for $\mathcal{P}$ may not be of much use if one hopes to identify the 'true' underlying tree and there are exponentially many such supertrees. It is intuitive to say that if there is a large number of consistent rooted binary supertrees for $\mathcal{P}$, then $\mathcal{P}$ doesn't contain much information about the 'true' tree. On the other hand, if there are only a few such supertrees, then $\mathcal{P}$ contains a lot of information about the 'true' tree. To be precise, suppose that the 'true' tree T on n labels is *a priori* equally likely to be any rooted binary phylogenetic tree on the n labels. Then the information about T given by $\mathcal{P}$ is

$$I(T|\mathcal{P}) = H(T) - H(T|\mathcal{P}) = -\log_2\left(\frac{N(\mathcal{P})}{(2n-3)!!}\right),$$

where H is the entropy function, $N(\mathcal{P})$ is the number of rooted binary phylogenetic trees consistent with $\mathcal{P}$, and $(2n-3)!!$ is the number of rooted binary phylogenetic trees on n labels. Consequently, counting the number of consistent rooted binary supertrees for $\mathcal{P}$ is a natural and realistic problem in phylogenetics. Unfortunately, the main result of this paper shows that this problem is computationally hard, in particular, #P-complete. An almost immediate corollary of the main result is that if $\mathcal{P}$ is a collection of unrooted binary phylogenetic trees, then counting the number of consistent (unrooted) binary supertrees for $\mathcal{P}$ is also #P-complete. This last result is not surprising as the associated decision problem is NP-complete [10].

The complexity class #P was introduced by Valiant [11] as an extension of classical complexity theory from decision problems to enumeration problems. The fact that computing the number of supertrees preserving a given set of relationships is #P-complete means that computing this number is as hard as computing any problem in the class #P. Such problems include counting the number of satisfying assignments to a Boolean formula in conjunctive normal form and counting the number of Hamiltonian circuits in a graph. Intuitively, this implies that it is extremely unlikely that there

exists a polynomial-time algorithm for computing the number of such supertrees. Indeed, such an algorithm would not only imply that P=NP, but that the whole 'polynomial hierarchy' collapses. For a good introduction to the complexity of counting problems, we refer the reader to Welsh [12].

2 Main Result

In this section, we formalise and state the main result. A brief description of the organisation of the paper is given at the end of the section. Throughout the paper, the phylogenetic notation and terminology follows Semple and Steel [9].

A *rooted phylogenetic tree* T (*on* X) is a rooted tree with the following properties:

(i) every interior vertex has degree at least three except for the root which may have degree two;

(ii) the leaves of T are bijectively labelled with the elements of X.

The set X is called the *label set* of T. Since X bijectively labels the leaves of T, we shall often view X has the leaf set of T. A rooted phylogenetic tree is *binary* if, in addition, every interior vertex has degree three except for the root which has degree two. Two rooted binary phylogenetic trees are shown in Fig. 1. For a collection $\mathcal{P}$ of rooted phylogenetic trees, we denote by $\mathcal{L}(\mathcal{P})$ the set

$$\bigcup_{T \in \mathcal{P}} \mathcal{L}(T),$$

where, for all T, the set $\mathcal{L}(T)$ denotes the label set of T.

Let X' be a subset of X, and suppose that T and T' are rooted binary phylogenetic trees on X and X', respectively. Then T *displays* T' if, up to suppressing degree two vertices, T' is isomorphic to the minimal rooted phylogenetic subtree of T whose label set is X'. Note that this minimal subtree is necessarily binary. To illustrate, T displays T' in Fig. 1.

A collection $\mathcal{P}$ of rooted binary phylogenetic trees is *compatible* if there exists a rooted binary phylogenetic tree T that displays every tree in $\mathcal{P}$, in which case, we say that T *displays* $\mathcal{P}$. If we view $\mathcal{P}$ as a collection of evolutionary trees on overlapping sets of species, then T displaying $\mathcal{P}$ corresponds to T preserving all of the ancestral relationships described by the trees in $\mathcal{P}$; that is, T is consistent with $\mathcal{P}$.

For an arbitrary collection $\mathcal{P}$ of rooted binary phylogenetic trees, Aho *et al.* [1] presented a polynomial-time algorithm for deciding whether or not

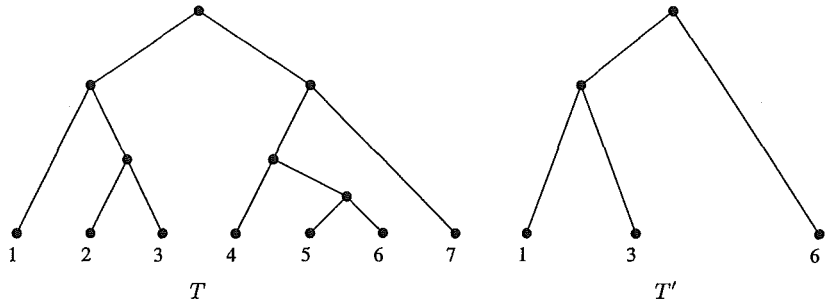

Figure 1: T displays T'.

$\mathcal{P}$ is compatible. Their algorithm, called BUILD, is constructive and, in the case $\mathcal{P}$ is compatible, one can obtain a rooted binary phylogenetic tree that displays $\mathcal{P}$ by refining the rooted phylogenetic tree outputted by BUILD. Furthermore, if BUILD outputs a rooted binary phylogenetic tree, then it is the unique rooted binary phylogenetic tree that displays $\mathcal{P}$. In contrast to these results, it is an immediate consequence of Theorem 2.1 below that, in general, counting the number of rooted binary phylogenetic trees that display $\mathcal{P}$ is hard.

A *rooted triple* is a rooted binary phylogenetic tree with three leaves. The rooted triple with leaves a, b, and c is denoted $ab|c$ if the path from a to b does not intersect the path from c to the root. Note that we make no distinction between $ab|c$ and $ba|c$. In Fig. 1, T' is the rooted triple $13|6$.

Theorem 2.1 shows that the following counting problem is computationally hard:

#CONSISTENT SUPERTREES
Instance: A collection $\mathcal{P}$ of rooted triples.
Question: How many rooted binary phylogenetic trees with label set $\mathcal{L}(\mathcal{P})$ display $\mathcal{P}$?

Theorem 2.1 *Computing* #CONSISTENT SUPERTREES *is* #P*-complete.*

Since a collection of rooted triples is a special type of collection of rooted binary phylogenetic trees, Theorem 2.1 implies that counting the number of consistent supertrees for an arbitrary collection of rooted binary phylogenetic trees is also #P-complete.

An almost immediate consequence of Theorem 2.1 is the analogous counting result for the unrooted setting. A *phylogenetic tree* T *(on* X*)* is an

unrooted tree with no degree-two vertices and whose leaves are bijectively labelled with the elements of X. In addition, T is *binary* if all of the interior vertices have degree three. A binary phylogenetic X-tree T *displays* a binary phylogenetic X'-tree T' if $X' \subseteq X$ and, up to suppressing degree two vertices, the minimal phylogenetic subtree of T whose labelled set is X' is isomorphic to T'. The notions of compatibility and consistency for collections of rooted binary phylogenetic trees extend to collections of binary phylogenetic trees in the obvious way.

A *quartet* is a binary phylogenetic tree with four leaves. The unrooted counterpart of #CONSISTENT SUPERTREES is the following:

#UNROOTED CONSISTENT SUPERTREES
Instance: A collection $\mathcal{P}$ of quartets.
Question: How many unrooted binary phylogenetic trees with label set $\mathcal{L}(\mathcal{P})$ display $\mathcal{P}$?

Let $\mathcal{P}$ be a collection of rooted triples and let x be an element not in $\mathcal{L}(\mathcal{P})$. For each $T \in \mathcal{P}$, let T_x be the quartet obtained from T by adjoining x to the root of T by an edge and then viewing the resulting tree as unrooted. Let $\mathcal{P}_x = \{T_x : T \in \mathcal{P}\}$. Thus $\mathcal{P}_x$ is a collection of quartets. It is easily seen that if T is a rooted binary phylogenetic tree that displays $\mathcal{P}$, then the binary phylogenetic tree obtained from T by adjoining x to the root of T by an edge and viewing the resulting tree as unrooted displays $\mathcal{P}_x$. Moreover, the converse also holds. Corollary 2.2 now follows from Theorem 2.1.

Corollary 2.2 *Computing* #UNROOTED CONSISTENT SUPERTREES *is* #P*-complete.*

Evidently, Corollary 2.2 implies that, for an arbitrary collection of binary phylogenetic trees, counting the number of consistent unrooted supertrees is #P-complete. We remark here that Steel [10] showed that determining if a collection of quartets, and thus more generally a collection of binary phylogenetic trees, is compatible is NP-complete. However, the complexity of the associated uniqueness problem is open and appears to be difficult.

The remainder of the paper is organised as follows. Although Theorem 2.1 is the main result, there are two closely related counting problems that also turn out to #P-complete; we describe these problems in Section 3. Section 4 consists of the proof of Theorem 2.1 and the last section consists of some final remarks.

We close this section with some further definitions. A *rooted caterpillar* is a rooted binary phylogenetic tree for which the subgraph induced by the

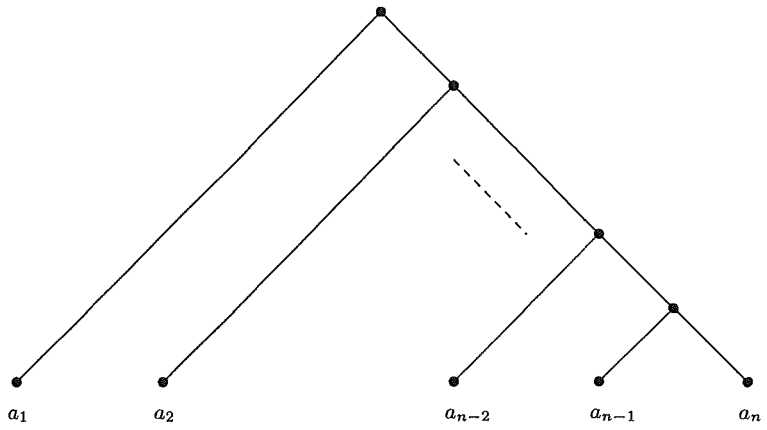

Figure 2: The rooted caterpillar $a_1a_2\cdots a_{n-2}|a_{n-1}a_n$.

set of interior vertices is a path. We denote the rooted caterpillar shown in Fig. 2 by $a_1a_2\cdots a_{n-2}|a_{n-1}a_n$.

Let T be a rooted phylogenetic tree. A useful partial order $\leq_T$ on the vertex set V of T is obtained by setting $u \leq_T v$ if the path from the root of T to v includes u. If $u \leq_T v$ and v is a leaf of T, we say that v is a *descendant label* of u. For all $a, b \in V$, we call the unique vertex of T that is the greatest lower bound of $\{a, b\}$ under $\leq_T$ the *most recent common ancestor* of a and b in T. Lastly, two distinct leaves of T form a *cherry* if they are adjacent to a common vertex.

3 Two Related Counting Problems

In this section, we describe two counting problems that are related to #Consistent Supertrees and which also turn out to be #P-complete. For each of these additional problems, all that is required is a relatively simple reduction from #Linear Extensions; that is, counting the number of linear extensions of a poset. This problem was shown to be #P-complete by Brightwell and Winkler [4].

As in the case of Theorem 2.1, all of the results in this section are stated in terms of collections of rooted triples, but each extends to collections of rooted binary phylogenetic trees. Furthermore, where the instance of a problem is a set $\mathcal{P}$ of rooted triples, we take $|\mathcal{P}|$ to be the number of labels present in $\mathcal{P}$. Since the number of distinct rooted triples on N labels is

$3\binom{N}{3}$, we can consider $|\mathcal{P}|$ to be our input size when examining polynomial reductions.

The first counting problem we consider is the following:

#CONSISTENT CATERPILLARS
Instance: A collection $\mathcal{P}$ of rooted triples.
Question: How many rooted caterpillars with label set $\mathcal{L}(\mathcal{P})$ display $\mathcal{P}$?

Structurally, rooted caterpillars are the simplest family of rooted binary phylogenetic trees. However, the next proposition shows that the above counting problem is hard.

Proposition 3.1 #CONSISTENT CATERPILLARS *is* #P*-complete.*

Proof. It is clear that #CONSISTENT CATERPILLARS is in #P since, given a rooted caterpillar T with label set $\mathcal{L}(\mathcal{P})$, one can verify whether T displays $\mathcal{P}$ in polynomial time.

Let $(\mathcal{S}, \prec)$ be a partially ordered set on n elements, and let x and y be distinct elements not in $\mathcal{S}$. Let $\mathcal{P}_{\mathcal{S}}$ be the following collection of rooted triples:

$$\{ax|b : a \prec b \in (\mathcal{S}, \prec)\} \cup \{xy|a : a \in \mathcal{S}\}.$$

Since the size of $\mathcal{P}_{\mathcal{S}}$ is polynomial in the input size of $(\mathcal{S}, \prec)$ and #LINEAR EXTENSIONS is #P-complete, to prove the proposition it suffices to show the number of linear extensions of $(\mathcal{S}, \prec)$ is equal to the number of rooted caterpillars with label set $\mathcal{L}(\mathcal{P})$ that display $\mathcal{P}$.

First suppose that $a_1 \prec a_2 \prec \cdots \prec a_n$ is a linear ordering of $\mathcal{S}$. Then $a_n a_{n-1} \cdots a_1 | xy$ is a rooted caterpillar that displays $\mathcal{P}_{\mathcal{S}}$. Moreover, this association of linear orderings of $(\mathcal{S}, \prec)$ with rooted caterpillars on $\{a_1, a_2, \ldots, a_n, x, y\}$ is one-to-one.

Now consider the other direction. As $xy|a \in \mathcal{P}_{\mathcal{S}}$ for all $a \in \mathcal{S}$, it is easily seen that $\{x, y\}$ is a cherry of any rooted binary phylogenetic tree and, in particular, any rooted caterpillar on $\{a_1, a_2, \ldots, a_n, x, y\}$ that displays $\mathcal{P}_{\mathcal{S}}$. It is now straightforward to check that if $b_n b_{n-1} \cdots b_1 | xy$ is a rooted caterpillar that displays $\mathcal{P}_{\mathcal{S}}$, where $b_1, b_2, \ldots, b_n \in \mathcal{S}$, then $b_1 \prec b_2 \prec \cdots \prec b_n$ is a linear extension of $(\mathcal{S}, \prec)$. As in the previous paragraph, this association is one-to-one. Therefore it follows that the number of linear extensions of $(\mathcal{S}, \prec)$ is equal to the number of rooted caterpillars with label set $\mathcal{L}(\mathcal{P}_{\mathcal{S}})$ that display $\mathcal{P}_{\mathcal{S}}$. □

Remark. Since all of the rooted triples in $\mathcal{P}_{\mathcal{S}}$ in the proof of Proposition 3.1 contain a common label, it follows that #CONSISTENT CATERPILLARS is

#P-complete even if all of the rooted triples in the input collection have a common label. This contrasts with the NP-complete decision problem of determining if a collection $\mathcal{P}$ of quartets is compatible. If each of the quartets in $\mathcal{P}$ share a common label, then the problem reduces to determining if an associated set of rooted triples is compatible, which can be done in polynomial time.

The second counting problem we consider in this section is the following:

#FORBIDDEN SUPERTREES
Instance: A collection $\overline{\mathcal{P}}$ of rooted triples.
Question: How many rooted binary phylogenetic trees T with label set $\mathcal{L}(\overline{\mathcal{P}})$ have the property that no rooted triple in $\overline{\mathcal{P}}$ is displayed by T?

Bryant [5] showed that the associated decision problem is NP-complete. Proposition 3.2 shows that #FORBIDDEN SUPERTREES is also hard.

Proposition 3.2 #FORBIDDEN SUPERTREES *is* #P*-complete.*

Proof. Given a rooted binary phylogenetic tree T with label set $\mathcal{L}(\overline{\mathcal{P}})$, it is clear that one can verify in polynomial time that no rooted triple in $\overline{\mathcal{P}}$ is displayed by T. Thus #FORBIDDEN SUPERTREES is in #P.

As in the proof of Proposition 3.1, the #P-complete problem we use for reduction is #LINEAR EXTENSIONS. Let $(\mathcal{S}, \prec)$ be a partially ordered set and let x be an element not in $\mathcal{S}$. Let $\overline{\mathcal{P}}_{\mathcal{S}}$ denote the following collection of rooted triples:

$$\{bx|a : a \prec b \in (\mathcal{S}, \prec)\} \cup \{ab|x : a, b \in \mathcal{S}\}.$$

Clearly, the size of $\overline{\mathcal{P}}_{\mathcal{S}}$ is polynomial in the imput size of $(\mathcal{S}, \prec)$. We complete the proof by showing that the number of linear extensions of $(\mathcal{S}, \prec)$ is equal to the number of rooted binary phylogenetic trees T with label set $\mathcal{L}(\overline{\mathcal{P}}_{\mathcal{S}})$ in which no rooted triple of $\overline{\mathcal{P}}_{\mathcal{S}}$ is displayed by T. In the latter, it turns out that all such trees are caterpillars.

Let $n = |\mathcal{S}|$. Suppose $a_1 \prec a_2 \prec \cdots \prec a_n$ is a linear ordering of $\mathcal{S}$. Then no rooted triple of $\overline{\mathcal{P}}_{\mathcal{S}}$ is displayed by the rooted caterpillar $a_n a_{n-1} \cdots a_2 | a_1 x$. It follows that, for each linear extension of $(\mathcal{S}, \prec)$, there is a distinct rooted caterpillar with the desired properties.

Now suppose that T is a rooted binary phylogenetic tree with label set $\mathcal{L}(\overline{\mathcal{P}}_{\mathcal{S}})$ and which has the property that no rooted triple of $\overline{\mathcal{P}}_{\mathcal{S}}$ is displayed by T. Since T does not display the rooted triple $ab|x$ for all distinct $a, b \in \mathcal{S}$, it follows that T has exactly one cherry and this cherry must contain

x. Consequently, T is a rooted caterpillar of the form $b_n b_{n-1} \cdots b_2 | b_1 x$. Suppose, for some $1 \leq i < j \leq n$, we have $b_j \prec b_i \in (\mathcal{S}, \prec)$. Then T displays $b_i x | b_j$ and $b_i x | b_j \in \overline{\mathcal{P}}_{\mathcal{S}}$, giving a contradiction. Hence, for each such rooted caterpillar, $b_1 \prec b_2 \prec \cdots \prec b_n$ is a linear extension of $(\mathcal{S}, \prec)$. As each rooted caterpillar induces a distinct linear extension, we deduce that the number of linear extensions of $(\mathcal{S}, \prec)$ is equal to the number of rooted binary phylogenetic trees T with label set $\mathcal{L}(\overline{\mathcal{P}}_{\mathcal{S}})$ in which no rooted triple of $\overline{\mathcal{P}}_{\mathcal{S}}$ is displayed by T. □

4 #Consistent Supertrees is #P-complete

This section consists of the proof of Theorem 2.1. The overall strategy of the proof follows Brightwell and Winkler's proof that #Linear Extensions is #P-complete [4]. One difference is that, for convenience, we use a reduction from #Mon-2-Sat instead of #3-Sat. The problem #Mon-2-Sat is in Valiant's original list of #P-complete functions [11] and is the problem of counting the number of satisfying assignments of a Boolean formula in conjunctive normal form that has exactly two literals per clause neither of which are negations. We remark here that, despite the reductions of the last section, it seems that there is no straightforward reduction from #Linear Extensions to #Consistent Supertrees.

Before presenting the proof, we give a brief outline of the general approach. Evidently, #Consistent Supertrees is in #P. Let I be a general instance of #Mon-2-Sat. The strategy is to choose a suitable set S of primes and, for each $p \in S$, convert I into a particular set $\mathcal{P}_I(p)$ of rooted triples so that the number (mod p) of rooted binary phylogenetic trees displaying $\mathcal{P}_I(p)$ is a simple multiple of the number of satisfying assignments of I. Using an oracle $\mathcal{O}(\mathcal{P})$ that can count the number of rooted phylogenetic trees that display a collection $\mathcal{P}$ of rooted triples in polynomial time, we can determine the number (mod p) of satisfying assignments of I for each $p \in S$. Because S is suitably chosen, we are then able to apply the Chinese Remainder Theorem and Euclid's Algorithm to recover the number of satisfying assignments of I exactly.

In the proof, we make use of the following two lemmas (see [4] and [9], respectively). The second lemma is freely used.

Lemma 4.1 *Let m be a positive integer. Then the product of the set of primes between $8m$ and $64m^2$ is at least $(8m)!2^{8m}$.*

For a positive odd number $2k+1$, we denote by $(2k+1)!!$ the product

$$(2k+1)!! = (2k+1) \times (2k-1) \times \cdots \times 3 \times 1.$$

Lemma 4.2 *Let $k \geq 2$. Then*

(i) the number of edges in any rooted binary phylogenetic tree on k labels is $(2k-2)$; and

(ii) the number of distinct rooted binary phylogenetic trees on k labels is $(2k-3)!!$.

To begin the formal proof, let I be an instance of #MON-2-SAT on n literals and m clauses. Without loss of generality, we may assume $m > n$, as we can always pad I with repeated clauses. Throughout the proof, the sets $\mathcal{P}$ of rooted triples we construct contain no more than $(8m)^3$ labels and so the input to the oracle $\mathcal{O}(\mathcal{P})$ is polynomially bounded. We denote the number of rooted binary phylogenetic trees displaying $\mathcal{P}$ by $N(\mathcal{P})$. Lastly, to ease reading, throughout the proof we write 'phylogenetic tree' for 'rooted binary phylogenetic tree'.

4.1 Determining the set S of primes

We first define a set $\mathcal{P}_I$ of rooted triples that will play an important role later in the proof. The set S of primes is chosen so that no member divides the number of phylogenetic trees displaying $\mathcal{P}_I$. The set of labels of $\mathcal{P}_I$ is

$$\{x_i : 1 \leq i \leq n\} \cup \{c_j^1, c_j^2, c_j^3 : 1 \leq j \leq m\} \cup \{b\}.$$

For each clause $c_j = (x_i \text{ or } x_k)$ of I, we include the following rooted triples in $\mathcal{P}_I$:

$$bx_i|c_j^1, \quad bx_k|c_j^1, \quad bx_i|c_j^2, \quad bx_k|c_j^3.$$

There are no other rooted triples in $\mathcal{P}_I$. Since $|\mathcal{L}(\mathcal{P}_I)| = n + 3m + 1$, there are at most $(2n+6m-1)!!$ distinct phylogenetic trees that display $\mathcal{P}_I$. In particular, as $n < m$, we have $N(\mathcal{P}_I) \leq (2n+6m-1)!! \leq (8m)!$. Let S_0 be the set of primes between $8m$ and $64m^2$. By Lemma 4.1, the product of the elements of S_0 is at least $(8m)!2^{8m}$. Since $N(\mathcal{P}_I) \leq (8m)!$, there is a subset S of S_0 with the properties that no element divides $N(\mathcal{P}_I)$ and the product of the elements is at least 2^{8m}. Since our input size is at least m, we can compute this set of primes in polynomial time.

4.2 Defining the set $\mathcal{P}_I(p)$ of rooted triples

In this subsection, we define a set $\mathcal{P}_I(p)$ of rooted triples for each prime p in S. For simplicity, we include several rooted caterpillars in $\mathcal{P}_I(p)$. Each such caterpillar $d_1 d_2 \cdots d_{k-2}|d_{k-1}d_k$ on k labels is simply replacing the set of triples

$$\{d_1|d_2d_3,\ d_2|d_3d_4,\ \ldots,\ d_{k-2}|d_{k-1}d_k\}$$

which defines it (see [9] for details).

Let $x_1, x_2, \ldots, x_n$ denote the n variables of I and, for notational convenience, let $c_{n+1}, c_{n+2}, \ldots, c_{n+m}$ denote the m clauses of I. The label set of $\mathcal{P}_I(p)$ is the union of the sets

$$\{x_i, \overline{x}_i : 1 \leq i \leq n\} \cup \{c_i^1, c_i^2, c_i^3, c_i^4 : n+1 \leq i \leq n+m\} \cup \{b_0\},$$
$$\{a_{i,1}, a_{i,2}, \ldots, a_{i,\frac{p+1}{2}}, h_{i,1}, h_{i,2}, \ldots, h_{i,\frac{p+1}{2}} : 1 \leq i \leq n+m\},$$
$$\text{and } \{u_{i,1}, u_{i,2}, \ldots, u_{i,p-2} : 1 \leq i \leq n+m\}.$$

Essentially, for all $n+1 \leq i \leq n+m$, the labels $c_i^1, c_i^2, c_i^3, c_i^4$ correspond to the four possible assignments to the variables in the clause c_i. We call the elements of $\{x_i, \overline{x}_i : 1 \leq i \leq n\}$ and $\{c_i^1, c_i^2, c_i^3, c_i^4 : n+1 \leq i \leq n+m\}$ the *literal* and *clause labels*, respectively.

We now describe the rooted triples of $\mathcal{P}_I(p)$; the label b_0 plays a special role in this description. We begin with those rooted triples not involving literal or clause labels. Firstly, for each i, $\mathcal{P}_I(p)$ contains the rooted caterpillar

$$b_0 a_{i,1} a_{i,2} \cdots a_{i,\frac{p-3}{2}} | a_{i,\frac{p-1}{2}} a_{i,\frac{p+1}{2}}.$$

In addition to these rooted caterpillars, $\mathcal{P}_I(p)$ also contains the rooted caterpillar

$$a_{n+m,1} a_{n+m-1,1} \cdots a_{2,1} | a_{1,1} b_0.$$

The fact that $\mathcal{P}_I(p)$ contains these $n+m+1$ rooted caterpillars means that any phylogenetic tree that displays $\mathcal{P}_I(p)$ must display the phylogenetic tree (solid lines) shown in Fig. 3. For any such phylogenetic tree T, let b_i denote the most recent common ancestor of b_0 and $a_{i,1}$, for all i.

Secondly, $\mathcal{P}_I(p)$ contains the rooted triples in the sets

$$\{b_0 a_{i-1,1} | h_{i,k} : 2 \leq i \leq n+m, 1 \leq k \leq \tfrac{p+1}{2}\},$$
$$\{b_0 a_{i-1,1} | u_{i,k} : 2 \leq i \leq n+m, 1 \leq k \leq p-2\},$$
$$\{b_0 h_{i,k} | a_{i,1} \quad : 1 \leq i \leq n+m, 1 \leq k \leq \tfrac{p+1}{2}\},$$
$$\text{and } \{b_0 u_{i,k} | a_{i,1} \quad : 1 \leq i \leq n+m, 1 \leq k \leq p-2\}.$$

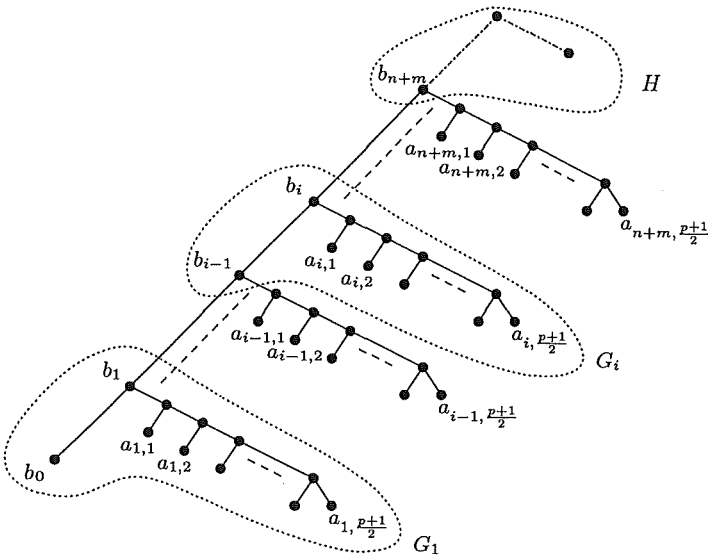

Figure 3: Any phylogenetic tree displaying $\mathcal{P}_I(p)$ displays the phylogenetic tree indicated by the solid lines.

Loosely speaking, for any phylogenetic tree displaying $\mathcal{P}_I(p)$, the above union of rooted triples forces the labels $h_{i,*}$ and $u_{i,*}$ to be 'sandwiched between' b_{i-1} and b_i for all i.

Lastly, we describe the rooted triples of $\mathcal{P}_I(p)$ that involve literal and clause labels. For each clause $c_i = (x_j \text{ or } x_k)$ of I, the set $\mathcal{P}_I(p)$ contains the sets

$$\{b_0x_j|c_i^1,\ b_0x_k|c_i^1,\ b_0x_j|c_i^2,\ b_0\overline{x}_k|c_i^2,\ b_0\overline{x}_j|c_i^3,\ b_0x_k|c_i^3,\ b_0\overline{x}_j|c_i^4,\ b_0\overline{x}_k|c_i^4\}.$$

Note that changing the order of the literals in the clause c_i only permutes the labels and gives rise to an isomorphic set of rooted triples. Finally, $\mathcal{P}_I(p)$ contains the sets

$$\{h_{i,1}h_{i,k}|x_i,\ h_{i,1}h_{i,k}|\overline{x}_i : 1 \leq i \leq n, 2 \leq k \leq \tfrac{p+1}{2}\}$$
$$\text{and } \{h_{i,1}h_{i,k}|c_i^l : n+1 \leq i \leq n+m, 2 \leq k \leq \tfrac{p+1}{2}, 2 \leq l \leq 4\}.$$

We will see in Section 4.4 that, under certain assumptions, the last two sets of rooted triples force exactly one of $x_i, \overline{x}_i$ or one of c_i^2, c_i^3, c_i^4 to be between b_{i-1} and b_i in any phylogenetic tree that displays $\mathcal{P}_I(p)$. The other literal

and clause labels are thus forced into 'the top part of the tree' indicated by dash-dot lines in Fig. 3.

4.3 Breaking the tree into sections

Fixing p, let T be a phylogenetic tree that displays $\mathcal{P}_I(p)$. Using the fact that T displays the phylogenetic tree shown in Fig. 3, we unambigously 'break' T into $n+m+1$ distinct sections $G_1(T), G_2(T), \ldots, G_{n+m}(T)$, and $H(T)$ as follows. For $1 \leq i \leq n+m$, let G_i be the phylogenetic subtree induced by b_i and its descendants with the descendants of b_{i-1} contracted to the single leaf b_{i-1}. In addition, let $H(T)$ denote the phylogenetic subtree obtained from T by contracting the descendants of b_{n+m} into the single leaf b_{n+m} (see Fig. 3).

We place an equivalence relation $\sim$ on the set of phylogenetic trees displaying $\mathcal{P}_I(p)$ by setting

$$T_1 \sim T_2 \iff \mathcal{L}(G_i(T_1)) = \mathcal{L}(G_i(T_2)) \quad \text{for all } 1 \leq i \leq n+m.$$

Note that the only labels of $\mathcal{P}_I(p)$ that are not forced to be in a specific $G_i(T)$ for every phylogenetic tree T displaying $\mathcal{P}_I(p)$ are the literal and clause labels.

Let ψ be an equivalence class under $\sim$. For all i, we denote the set $\mathcal{L}(G_i(T))$ by ψ_i and the set $\mathcal{L}(H(T))$ by ψ_H, where T is any phylogenetic tree in ψ. Furthermore, for all i, let $\mathcal{P}|\psi_i$ denote the set of rooted triples on ψ_i that is obtained by taking

(i) those rooted triples in $\mathcal{P}_I(p)$ all of whose labels are in ψ_i and

(ii) those rooted triples in $\mathcal{P}_I(p)$ that contain two distinct labels in ψ_i and one label in ψ_j, for some $j < i$, and replacing the label in ψ_j with the label b_{i-1}.

The set $\mathcal{P}|\psi_H$ of rooted triples on ψ_H is similarly defined.

Now let T be a phylogenetic tree in ψ. Then it is clear that, for all i, the phylogenetic subtrees $G_i(T)$ and $H(T)$ display $\mathcal{P}|\psi_i$ and $\mathcal{P}|\psi_H$, respectively. Furthermore, in each $G_i(T)$, the most recent common ancestor of b_{i-1} and $a_{i,1}$ is the root of $G_i(T)$. We will say that a phylogenetic tree G_i is *good* for $\mathcal{P}|\psi_i$ if G_i displays $\mathcal{P}|\psi_i$ and the most recent common ancestor of b_{i-1} and $a_{i,1}$ is the root of G_i. Thus if, for all i, G_i is a phylogenetic tree that is good for $\mathcal{P}|\psi_i$ and H is a phylogenetic tree that displays $\mathcal{P}|\psi_H$, then a unique phylogenetic tree in ψ is obtained by joining these trees through the

vertices $b_1, b_2, \ldots, b_{n+m}$. It now follows that

$$|\psi| = N(\mathcal{P}|\psi_H) \prod_{i=1}^{n+m} N'(\mathcal{P}|\psi_i), \tag{1}$$

where $N'(\mathcal{P}|\psi_i)$ is the number of phylogenetic trees on label set ψ_i which are good for $\mathcal{P}|\psi_i$. We will show that $|\psi| \not\equiv 0 \pmod{p}$ if and only if ψ corresponds to a satisfying assignment of I.

4.4 Isolating the labels

Suppose that $|\psi| \not\equiv 0 \pmod{p}$. Then, for all $1 \leq i \leq n+m$, it follows by (1) that

$$N'(\mathcal{P}|\psi_i) \not\equiv 0 \pmod{p}.$$

We first show that, for each i, $N'(\mathcal{P}|\psi_i) \not\equiv 0 \pmod{p}$ if and only if ψ_i contains exactly one of the literal labels $x_i, \overline{x}_i$ if $i \leq n$ and exactly one of the clause labels c_i^2, c_i^3, c_i^4 if $i \geq n+1$, but no other literal or clause labels.

Fixing i, let $\mathcal{A}$ be the set of literal and clause labels contained in ψ_i. Every phylogenetic tree that is good for $\mathcal{P}|\psi_i$ has the $h_{i,*}$'s, $u_{i,*}$'s and b_{i-1} on one side of the root, and the $a_{i,*}$'s on the other side. Hence such a tree partitions $\mathcal{A}$ into two parts $\mathcal{A}^1$ and $\mathcal{A}^2$ depending upon whether an element of $\mathcal{A}$ is on the same side of the root as b_{i-1} or $a_{i,1}$, respectively. Now no rooted triple of $\mathcal{P}_I(p)$ containing a literal or clause label also contains an 'a' label. Consequently, for a particular partition $\{\mathcal{A}^1, \mathcal{A}^2\}$, the number of phylogenetic trees that induce this partition is divisible by the number of distinct ways of attaching the labels of $\mathcal{A}^2$ to the side of the root containing $a_{i,1}$. If $\mathcal{A}^2$ is non-empty, then, as there are initially p available edges on this side of the root, the number of such trees is certainly divisible by p and therefore contributes zero to $N'(\mathcal{P}|\psi_i) \pmod{p}$. Thus, in counting $\pmod{p}$, we need only consider those phylogenetic trees displaying $\mathcal{P}|\psi_i$ for which $\mathcal{A}^2$ is empty.

Let $|\mathcal{A}| = k$. The number $\pmod{p}$ of phylogenetic trees that are good for $\mathcal{P}|\psi_i$ is equal to the number $\pmod{p}$ of phylogenetic trees that can be constructed by first taking a phylogenetic tree T on all of the labels except the $u_{i,*}$'s, and then attaching the $u_{i,*}$'s. Since the only rooted triple in $\mathcal{P}|\psi_i$ that has $u_{i,j}$ as a label is $b_{i-1}u_{i,j}|a_{i,1}$, it follows that $u_{i,j}$ can be attached to any edge on the side of the root containing b_{i-1}. Since we need only consider those phylogenetic trees for which $\mathcal{A}^2$ is empty, there are

$$\frac{p+1}{2} + k + 1$$

labels on this side of the root before attaching the $u_{i,*}$'s. Therefore there are $p+2k+2$ available edges. Thus there are $p+2k+2$ ways of attaching $u_{i,1}$. As attaching $u_{i,1}$ creats two additional edges, there is now $p+2(k+2)$ ways of attaching $u_{i,2}$. Continuing this process, we eventually have $p+2(k+p-2)$ ways of attaching $u_{i,p-2}$. Hence there are

$$(p+2k+2)(p+2k+4)\cdots(3p+2k-4)$$

distinct ways of attaching the $u_{i,*}$'s to T. Since $k \leq 2n+4m < 8m < p$, this product is zero (mod p) unless $k=0$ or $k=1$, therefore if $|\mathcal{A}| > 1$ then $N'(\mathcal{P}|\psi_i) \equiv 0 \pmod{p}$.

Now suppose that $|\mathcal{A}| \leq 1$. Furthermore, suppose that neither x_i nor $\overline{x}_i$ is in $\mathcal{A}$ if $i \leq n$, or that none of c_i^2, c_i^3, c_i^4 is in $\mathcal{A}$ if $i \geq n+1$. Then the only rooted triple in $\mathcal{P}|\psi_i$ that has $h_{i,j}$ as a label is $b_{i-1}h_{i,j}|a_{i,1}$. Since we may assume that $|\mathcal{A}^2| = 0$, this implies that there are

$$\frac{p+1}{2} + (p-2) + 1 + k = \frac{3p-1}{2} + k$$

labels on the side of the root containing b_{i-1} which are otherwise unconstrained. Therefore the number of phylogenetic trees that are good for $\mathcal{P}|\psi_i$ is divisible by $(3p+2k-4)!!$ and thus divisible by p. Hence, under these conditions, $N'(\mathcal{P}|\psi_G) \equiv 0 \pmod{p}$. It follows that if $N'(\mathcal{P}|\psi_i) \not\equiv 0 \pmod{p}$, then $|\mathcal{A}| = 1$ and $\mathcal{A}$ consists of exactly one of $x_i, \overline{x}_i$ if $i \leq n$, or one of c_i^2, c_i^3, c_i^4 if $i \geq n+1$. Next we show that, in this circumstance,

$$N'(\mathcal{P}|\psi_i) \equiv \frac{(3p-2)!!}{p} \not\equiv 0 \pmod{p}.$$

First suppose that $i \leq n$ and $\mathcal{A}$ consists of exactly one element of $x_i, \overline{x}_i$. Without loss of generality, we may assume that $\mathcal{A} = \{x_i\}$. If x_i is on the same side of the root as $a_{i,1}$, then arguing as previously the number of phylogenetic trees that are good for $\mathcal{P}|\psi_i$ and have this property contribute zero (mod p) to the sum. Hence we may assume that x_i is on the side of the root containing b_{i-1}. It now follows that the side of the root containing $a_{i,1}$ is fixed, and so it suffices to count the number of phylogenetic trees on the label set

$$\{x_i, b_{i-1}\} \cup \{h_{i,j} : 1 \leq j \leq \tfrac{p+1}{2}\} \cup \{u_{i,j} : 1 \leq j \leq p-2\}.$$

that display the triples in the set

$$\{h_{i,1}h_{i,j}|x_i : 2 \leq j \leq \tfrac{p+1}{2}\}.$$

Now the number of phylogenetic trees on $\{x_i\} \cup \{h_{i,j} : 1 \leq j \leq \frac{p+1}{2}\}$ that display these triples is $(p-2)!!$. As $b_{i-1}, u_{i,1}, u_{i,2}, \ldots, u_{i,p-2}$ are unconstrained, it follows that the number of phylogenetic trees on $\{x_i, b_{i-1}\} \cup \{h_{i,j} : 1 \leq j \leq \frac{p+1}{2}\} \cup \{u_{i,j} : 1 \leq j \leq p-2\}$ that display these triples is

$$(p-2)!!(p+2)(p+4)\cdots(3p-2).$$

Hence, if $i \leq n$ and $\mathcal{A}$ consists of one element from $x_i, \overline{x}_i$, then

$$N'(\mathcal{P}|\psi_i) \equiv \frac{(3p-2)!!}{p} \not\equiv 0 \pmod{p}. \tag{2}$$

Similarly, if $i \geq n+1$ and $\mathcal{A}$ consists of one element from c_i^2, c_i^3, c_i^4, then

$$N'(\mathcal{P}|\psi_i) \equiv \frac{(3p-2)!!}{p} \not\equiv 0 \pmod{p}. \tag{3}$$

4.5 Determining the number of solutions of I

Suppose that $|\psi| \not\equiv 0 \pmod{p}$. Then, by the last subsection, each ψ_i contains exactly one of the literal labels $x_i, \overline{x}_i$ if $i \leq n$, or exactly one of the clause labels c_i^2, c_i^3, c_i^4 if $i \geq n+1$. Thus the remaining literal and clause labels are in ψ_H. In particular, $c_i^1 \in \psi_H$ for $n+1 \leq i \leq n+m$. Under the assumption $|\psi| \not\equiv 0 \pmod{p}$, the literal labels appearing in ψ_H correspond to a satisfying truth assignment of I as follows. Consider the truth assignment given by setting x_i true if $x_i \in \psi_H$ and false if $\overline{x}_i \in \psi_H$. To see this is a satisfying assignment for I, suppose that some clause $c_i = (x_j \text{ or } x_k)$ in I is not satisfied. Then $\overline{x}_j, \overline{x}_k \in \psi_H$, and so, as $\mathcal{P}_I(p)$ contains the rooted triples

$$b_0\overline{x}_k|c_i^2,\ b_0\overline{x}_j|c_i^3,\ b_0\overline{x}_j|c_i^4,$$

we must have $c_i^2, c_i^3, c_i^4 \in \psi_H$. But then $\psi_i \cap \{c_i^2, c_i^3, c_i^4\} = \emptyset$, contradicting the assumption that $|\psi| \not\equiv 0 \pmod{p}$.

Similarly, a satisfying assignment for I gives rise to a unique equivalence class ψ as follows. For each true literal, we assign x_i to ψ_H and $\overline{x}_i$ to ψ_i and, for each false literal, we assign $\overline{x}_i$ to ψ_H and x_i to ψ_i. For each clause $c_i = (x_j \text{ or } x_k)$, we place the clause label that is related only to false versions of the literals in ψ_i and the rest of the clause labels in ψ_H. Since the assignment is satisfying, c_i^1 is in ψ_H. Thus a satisfying assignment defines an equivalence class ψ, and the analysis of Section 4.4 implies that $|\psi| \not\equiv 0 \pmod{p}$.

Now suppose ψ is an equivalence class corresponding to a satisfying assignment for I. Then the set $\mathcal{P}|\psi_H$ of rooted triples is isomorphic to the set $\mathcal{P}_I$ of rooted triples defined in Section 4.1. Hence by (1), (2), and (3),

$$|\psi| \equiv N(\mathcal{P}_I)\left(\frac{(3p-2)!!}{p}\right)^{n+m} \pmod{p}.$$

Since we have chosen p from a set of primes none of which divide $N(\mathcal{P}_I)$,

$$N(\mathcal{P}_I)\left(\frac{(3p-2)!!}{p}\right)^{n+m} \not\equiv 0 \pmod{p}.$$

Thus, for any equivalence class ψ, $|\psi| \pmod{p}$ is either zero (if ψ does not correspond to a satisfying assignment) or a constant depending only on n, m, p, and $N(\mathcal{P}_I)$. Let $s(I)$ be the number of satisfying assignments of I. Then

$$N(\mathcal{P}_I(p)) \equiv N(\mathcal{P}_I)\left(\frac{(3p-2)!!}{p}\right)^{n+m} s(I) \pmod{p}.$$

We determine $N(\mathcal{P}_I(p))$ for each $p \in S$ and $N(\mathcal{P}_I)$ using $|S|+1$ oracle calls. Note that there are

$$\begin{aligned} n(2p-1) + m(2p-1) + 2n + 4m + 1 &\leq 2m(2p+2) + 1 \\ &< 2m(128m^2 + 2) + 1 \\ &< (8m)^3 \end{aligned}$$

labels in $\mathcal{P}_I(p)$ and at most $4m$ labels in $\mathcal{P}_I$, and so we can legitimately use the oracle to determine the number of phylogenetic trees displaying these sets. Thus, for each $p \in S$, we can determine $s(I) \pmod{p}$ in polynomial time. Since the product of the primes in the set S is at least 2^{8m} which is greater than 2^n and $s(I)$ is at most 2^n, this uniquely determines $s(I)$ by the Chinese Remainder Theorem. The number of satisfying assignments for I can now be recovered exactly using Euclid's Algorithm. We conclude that #MON-2-SAT is reducible to #CONSISTENT SUPERTREES and thus the latter problem is #P-complete. □

5 A Final Remark

It is interesting to note that, although #CONSISTENT SUPERTREES is hard, there exists an algorithm that outputs a list of rooted binary phylogenetic trees that display $\mathcal{P}$ with the properties that no tree is repeated, each tree is

generated in polynomial time, and all trees are listed [6, 8]. Since the total number of rooted binary phylogenetic trees that display $\mathcal{P}$ can be exponentially large, the total running time of this algorithm may be exponential. The existence of a randomised algorithm that could generate a rooted binary phylogenetic tree in polynomial time such that the tree was selected uniformly at random from the set of all rooted binary phylogenetic trees that display $\mathcal{P}$ would yield an efficient method of approximating the number of consistent supertrees (an FPRAS, see [7] for further details). The natural next step is to try and determine whether such an algorithm exists.

6 Acknowledgments

We thank Dominic Welsh for providing helpful comments on an earlier draft of this paper.

The first author was funded by the EPSRC and Vodafone, and supported in part by the RAND-APX. The second author was supported by the New Zealand Marsden Fund. This research was conducted while the second author held a Canterbury Fellowship at the University of Oxford and Visiting Research Fellowship at Merton College.

References

[1] A. V. Aho, Y. Sagiv, T. G. Szymanski, J. D. Ullman, Inferring a tree from lowest common ancestors with an application to the optimization of relational expressions, SIAM Journal on Computing 10 (1981) 405–421.

[2] O. R. P. Bininda-Emonds, J. L. Gittleman, M. A. Steel, The (super)tree of life: procedures, problems and prospects, Annual Reviews of Ecology and Systematics 33 (2002) 265–289.

[3] O. R. P. Bininda-Emonds, Phylogenetic supertrees, in preparation.

[4] G. Brightwell, P. Winkler, Counting linear extensions, ORDER 8:3 (1992) 225–242.

[5] D. Bryant, Building trees, hunting for trees, and comparing trees: the theory and methods in phylogenetic analysis, Ph.D. thesis, University of Canterbury (1997).

[6] M. Constantinescu, D. Sankoff, An efficient algorithm for supertrees, Journal of Classification 12 (1995) 101–112.

[7] M. R. Jerrum, L. G. Valiant, V. V. Vazirani, Random generation of combinatorial structures from a uniform distribution, Theoretical Computer Science 43 (1986) 169–188.

[8] M. P. Ng, N. C. Wormald, Reconstruction of rooted trees from subtrees, Discrete Applied Mathematics 69 (1996) 19–31.

[9] C. Semple, M. Steel, Phylogenetics, Oxford University Press, 2003.

[10] M. Steel, The complexity of reconstructing trees from qualitative characters and subtrees, Journal of Classification 9 (1992) 91–116.

[11] L. G. Valiant, The complexity of enumeration and reliability problems, SIAM Journal on Computing 8 (1979) 410–421.

[12] D. Welsh, Complexity: Knots, Colourings and Counting, Cambridge University Press, 1993.